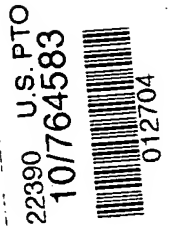


This Page Is Inserted by IFW Operations
and is not a part of the Official Record



BEST AVAILABLE IMAGES

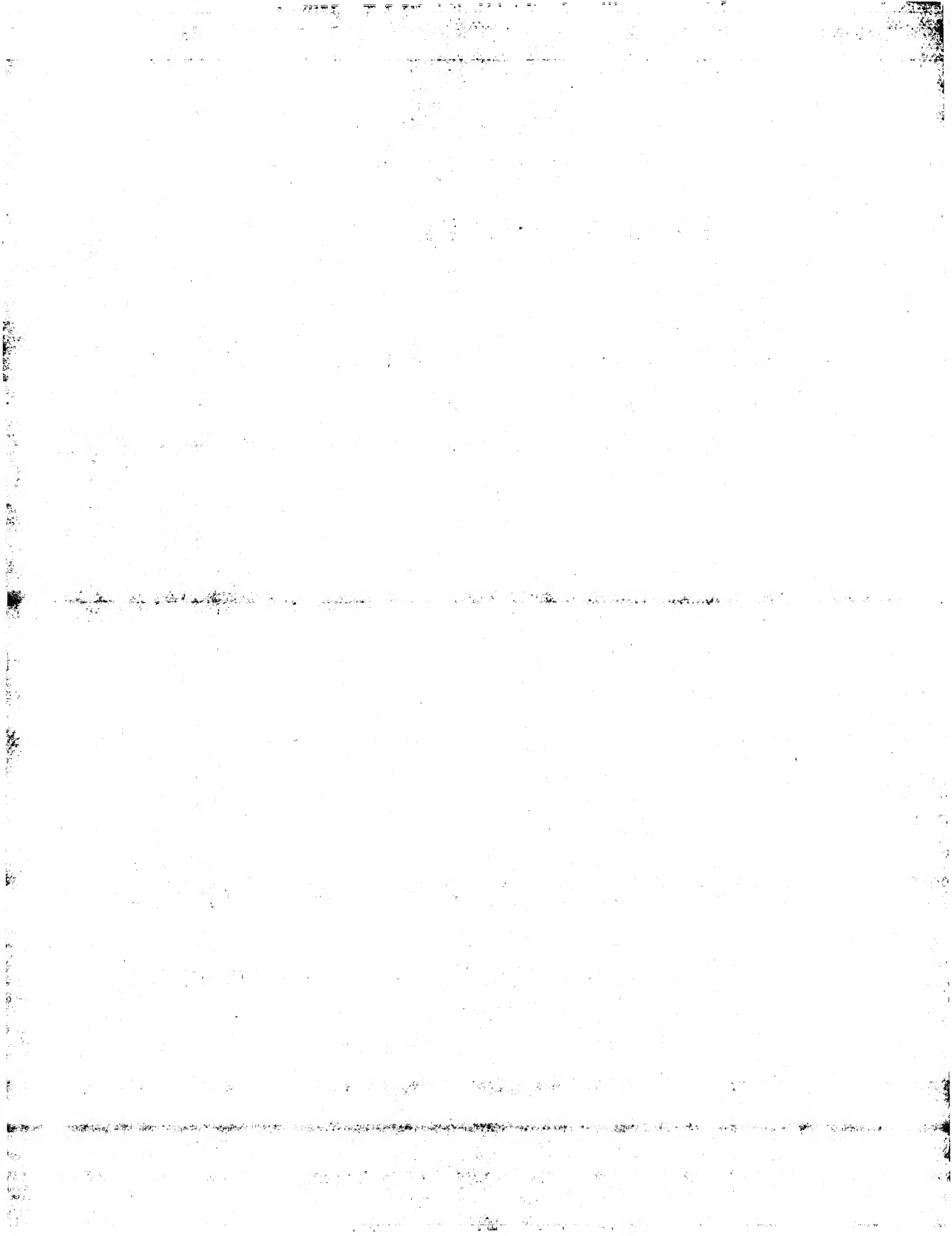
Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



**ARITHMETIC UNIT UTILIZING LOGARITHMIC EXPRESSION
NUMERICAL VALUE**

Patent Number: JP3014128
Publication date: 1991-01-22
Inventor(s): KUROKAWA TOMIO
Applicant(s): TOMIO KUROKAWA
Requested Patent: ☐ JP3014128
Application Number: JP19890151108 19890613
Priority Number(s):
IPC Classification: G06F7/00
EC Classification:
Equivalents:

Abstract

PURPOSE: To increase arithmetic speed, and also, to decrease an arithmetic error by converting input data to a logarithmic expression numerical value by using an input look-up table, inputting it to an arithmetic unit, converting the result of its operation to the number of fixed points by using an output look-up table if necessary and outputting it.

CONSTITUTION: Input data (fixed point data, etc.) is converted to a logarithmic expression numerical value by using an input look-up table 1, and inputted to an arithmetic unit 2 utilizing a logarithmic expression numerical value arithmetic method. Subsequently, an operation is executed by the arithmetic unit 2, and the result of its operation is converted to the number of fixed points or the number of floating points by using an output look-up table 3 if necessary. In such a manner, a special logarithmic analog amplifier and an exponential analog amplifier are not required, and by adding this arithmetic unit to the digital device of an existing computer, etc., the processing at high speed and with high accuracy is realized.

Data supplied from the esp@cenet database - I2

THIS PAGE BLANK (USPTO)

⑨ 日本国特許庁(JP) ⑩ 特許出願公開
⑪ 公開特許公報(A) 平3-14128

⑫ Int. Cl.⁷ 識別記号 庁内整理番号 ⑬ 公開 平成3年(1991)1月22日
G 06 F 7/00 7313-5B G 06 F 7/00 1 0 1 W
7313-5B T
審査請求 未請求 請求項の数 6 (全9頁)

⑭ 発明の名称 対数表現数値利用の演算装置

⑮ 特 願 平1-151108

⑯ 出 願 平1(1989)6月13日

⑰ 発 明 者 黒 河 富 夫 愛知県豊田市保見ヶ丘6丁目1番地1 公団110-403号
⑱ 出 願 人 黒 河 富 夫 愛知県豊田市保見ヶ丘6丁目1番地1 公団110-403号

明 細 書

1. 発明の名称

対数表現数値利用の演算装置

2. 特許請求の範囲

1. 対数表現数値演算法利用の演算装置(2)において、固定小数点数で表される入力データを入力ルックアップテーブル(1)を使用して対数表現数値に変換し、それを対数表現数値演算法利用の演算装置(2)への入力とし、その演算結果は必要に応じて出力ルックアップテーブル(3)を使用して、固定小数点数に変換して出力する演算装置

2. 移動小数点数を入力データとする入力ルックアップテーブル(1)とした、請求1の演算装置

3. 移動小数点数を出力データとする出力ルックアップテーブル(3)とした、請求1の演算装置

4. 移動小数点数を入力データとする入力ルックアップテーブル(1)とし、移動小数点数を出力

力する出力ルックアップテーブル(3)とした、

請求1の演算装置

5. 対数表現数値を入力データとするが、入力ルックアップテーブル(1)を経ないで対数表現数値演算法利用の演算装置(2)へ直接入力する、請求1の演算装置

6. 対数表現数値演算法利用の演算装置(2)の出力データを、出力ルックアップテーブル(3)を経ないで出力する、請求1の演算装置

3. 発明の詳細な説明

(産業上の利用分野)

コンピュータなどのディジタル装置において、数値データ、音声データ、メモリのアドレス値、画像配列上の画素値などディジタル数値(特に固定小数点数)で表現されるデータを高速に精度よく演算することに関する。

(従来の技術)

従来、固定小数点数で表されたデータは、固定小数点数のまま固定小数点演算法で、または移動小数点数に変換して移動小数点演算法で演算され

ていた。固定小数点演算の場合演算速度は速いが、演算誤差が大きく、また、ダイナミックレンジも小さく、演算の途中結果が大きくなる場合、数値が表現できない欠点がある。浮動小数点演算で行う場合、演算誤差は比較的小さいが、演算速度が遅く、演算精度も低くなる欠点がある。

また従来から、固定小数点演算法、浮動小数点演算法以外の演算法で、対数表現数値演算法 (Logarithmic Arithmetic) という高速で精度のよい演算法が知られている (例えば、文献: Kingsbury and Rayner, "Digital filtering using logarithmic arithmetic", Electron Lett., vol. 7, pp. 58-59, Feb. 1971)。その数値演算の原理を以下で示す。

対数表現数値演算法について:

対数表現数値演算法においては、数値は式 (1) のように表現される。

$$sd, d_1, d_2, \dots, d_n, f_1, f_2, \dots, f_n \quad (1)$$

ここで、 s, d_1, f_1 は 0 または 1 で s は数値の符号、 d -部と f -部の間には小数点があるとし、従って、 d -部と f -部の間には小数点があると、 d -部

(3)

次に演算であるが、 a から a' を算出する場合、答えを a' とすると、

$$a' = a \cdot a' = a \cdot (1 - a \cdot 10^{-n}) \quad (5)$$

となり、対数をとると、

$$z = x + \log(1 - a \cdot 10^{-n}) \quad (6)$$

となる。従って、 $\log(1 - a \cdot 10^{-n})$ をルックアップテーブルで予め用意しておけば、 z は高速に演算できる。 $x < y$ の場合は x と y を入れ替えた形で上記と同様にできる。 $x = y$ のときは算の値はゼロであるが、式 (2) はゼロを表現できないのでゼロに最も近い値を与える (ルックアップテーブルでそのように用意しておく)。従って、 z は負の数で表現可能な絶対値の最も大きい値となる。

数値演算も高速にできる。例えば、ある数 x の u 乗 (x^u , u は整数) の計算が非常に高速にできる。対数表現数値では指数を u 倍すればよい。簡単な例は平方と平方根の計算である。平方の計算は指数を 1 ビット左シフトするだけである。平方根の場合は指数を 1 ビット右シフトするだけである。

(5)

部は指数部、 f -部は小数部である。対数の値は略率の値で 1 より大きい数 a である。式 (1) は式 (2) の値を表す。

$$a = a' + a'' + a''' + \dots + a^{(n)} + a^{(n+1)} + \dots \quad (2)$$

従って、 d -部と f -部は小数部のある固定小数点数で数値の指数である。対数表現数値演算法においては、乗除算は指数の固定小数点演算の加減または減算である。従って、演算誤差はなく、高速にできる。加算を説明する。二つの数 a' と a'' の加算を考える。その和を a' とすると、 z を求めることが加算である。

$$a' = a' + a'' = a' (1 + a'' \cdot 10^{-n}) \quad (3)$$

であるから、 a を係とする対数をとると、式 (3) は

$$z = x + \log(1 + a'' \cdot 10^{-n}) \quad (x = y \text{ の場合}) \quad (4)$$

となる。 $\log(1 + a'' \cdot 10^{-n})$ は $1y = x1$ の関数であるから、 $1y = x1$ をアドレスとするルックアップテーブルにより高速に演算できる。従って、対数表現数値演算法における加算は高速にできる。 $x < y$ の場合は x と y を交換するだけで、同じように演算できる。

(4)

また、対数表現数値演算法は同条件 (同じ精度のダイナミックレンジと一定のビット数と同じ) の浮動小数点演算よりも演算精度が高いことが知られている (例えば、文献: Eusekava, Payne and Lee, "Error analysis of recursive digital filters implemented with logarithmic number systems", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-28, pp. 706-715, Dec. 1980)。

従来、対数表現数値演算法による処理システムは第 2 図で示す構成になっている。対数型アナログアンプ (4) は、アナログ信号をアナログのまま対数に変換する ($x = \log x$) 回路である。AD コンバータ (5) は通常の AD コンバータであり、アナログ信号をデジタル信号に変換する。対数表現数値演算法利用の演算回路 (2) は対数表現数値演算法による演算回路である。DA コンバータ (6) は通常の DA コンバータであり、デジタル信号をアナログ信号に変換する。数値型アナログアンプ (7) はアナログ信号をアナログのまま

(6)

倍率変換する($x \rightarrow x^a$)。

以上のごとき構成されるシステムでは、対数型アナログアンプ(4)や階数型アナログアンプ(7)という特別のハードウェアが必要であり、しかもその特性が対数表現数値演算法利用の演算装置(2)と適合していなければならない。また、既存の画像、音声、その他データ入力装置によるデータは、そのままでは対数表現数値演算法利用の演算装置(2)への入力データとはならない。しかも、対数表現数値演算法利用の演算装置(2)はデジタル・フィルタ等の限られた分野にしか使われていない。

(発明が解決しようとする課題)

本発明は、上記従来法の欠点を解消するもので、演算速度が速くて、演算誤差の小さい、しかも簡便な演算装置を提供することを目的とする。

(問題を解決するための手段)

第1図は本発明の構成を示す図である。入力データ(固定小数点データなど)を入力ルックアップテーブル(1)を使用して、対数表現数値に変換(7)

数が存在しているからである。例えば $m=4$, $n=10$, $a=2$ の場合を考える。これは18ビットの長さで丁度可能である。この場合第3図で説明すると、横軸に i があるが、 $i=1$ から i までの間には 2^{i-1} 個の数が有り($n=10$ であるから)、従って縦軸においては、 2^{i-1} から 2^i までの間には 2^{i-1} 個の数が有る。一般に数の密度が低いところは、255が N の上限の場合、254と255の間であるが、ここでも、その間には約6個の数が有る。99から100の間には約15個の数が存在する。20ビットの間長にして、 $m=4$, $n=14$, $a=2$ とすると、整数254と255の間には約92個の数が存在する。従って、あまり大きくない数 N であれば、相当程度よく対数表現数値に変換することが可能である。この変換はルックアップテーブルを使用すると高速にできる。すなわち、これが第1図の入力ルックアップテーブル(1)の働きである。

なお、上記の数 N は整数である必要はなく、小数部のある固定小数点数あるいは浮動小数点数で

(9)

換して(入力データが対数表現数値の場合は入力ルックアップテーブル(1)を越さない)、対数表現数値演算法利用の演算装置(2)へ入力し、そこで演算を行い、その演算結果は必要に応じて出力ルックアップテーブル(3)を使用して、固定小数点数または浮動小数点数に変換することにより、または変換しないで対数表現数値のまま出力することにより、演算を高速高精度に行うことができる。

(作用)

第3図は対数表現数値の階数(1:階数ではなく、式(1)の形式の固定小数点数)とそれによる a^i の関数($N=a^i$, $a=2$)を示す図である。限られたビット数(例えば8ビット)で表されるような整数 N (例えば0-255)は、式(1)での $m+n$ を比較的大きくとれば(例えば、 $m+n=14$ で、一語16ビット)底 a を適正に決めることにより相当に精度よく式(1)の形の対数表現数値に変換できる。すなわち N は a^i の形で近似できる。これは整数 N の近似には多くの a^i 形式の

(8)

もよい。いずれにしても、その数が表現されているビットパターンをアドレスとすれば、ルックアップテーブルを使用することにより、その数に応じた値を提供できる。

変換後の計算は対数表現数値演算法利用の演算装置(2)で行うのでこれも高速高精度にできる。

演算の結果を固定小数点数または浮動小数点数の形に直して出力する場合、出力の長さが必要に応じて長くすれば、精度は落ちない。固定小数点数の出力の場合も精度を要する場合、小数部のビット数を多くすればよい。これも出力ルックアップテーブル(3)で用意しておけばよい。従って、変換は高速にできる。

(発明の実施例)

実施例1

画像の幾何学的変換を第4、5図等を使用して説明する。第4図は画像の幾何学的変換を示す図である。すなわち、原画像(41)から幾何学的変換によって変換後の画像(42)を得た図である。式(7)は拡大、縮小、回転、平行移動等の

(10)

座標変換をするための座標変換式（アフィン変換の式）である。

$$X_1 = a_{11}X_0 + a_{12}Y_0 + b_1$$

$$Y_1 = a_{21}X_0 + a_{22}Y_0 + b_2$$

(7)

式(7)では (X_0, Y_0) は原画像(41)の画像平面上の画素の座標を渡し、 (X_1, Y_1) は変換後の画像(42)の画像平面上で、 (X_0, Y_0) に相当する画素の座標である。式(7)は式(8)のように書き換えることができる。

$$X_0 = b_{11}X_1 + b_{12}Y_1 + b_3$$

$$Y_0 = b_{21}X_1 + b_{22}Y_1 + b_4$$

(8)

式(8)は変換後の画像(42)の画像平面上の座標 (X_1, Y_1) の画素は原画像(41)の画像平面上のどの画素が相当しているかを表す。通常、 (X_1, Y_1) を整数座標として、 (X_0, Y_0) が丁度整数座標になることは殆どなく、計算される座標は小数部分を含んだ形になり、原画像(41)の画素に丁度対応することはない。第5図はその様子を示す図である。計算式による座標点 (X_0, Y_0) (51)は格子点(52)（整数座標）に一致することは

(11)

る。原画像(41)の画像平面上のその座標から変換後の画像(42)の画像平面上の座標 (X_1, Y_1) に画素を移す。上記のことを座標 (X_1, Y_1) について繰り返すことにより、アフィン変換が高速度高精度にできる。

実施例2

実施例1で示したようなアドレスの計算ではなくデータ自身の演算にも本発明は有効である。既存の画像処理装置は整数データを多く扱う。少なくとも、画像データの入力またはディスプレイ装置等への出力は整数（もう少し広く固定小数点数）の形で行う場合が多い。それは通常0-255の整数である。従って、実施例1で示した演算と同様に、その演算は高速度高精度に行うことができる。式(9)は2次元データ $g(m, n)$ のフーリエ変換式である。

$$G(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} g(m, n) V_1^m V_2^n \quad (9)$$

ただし、 $V_1 = \exp(-j2\pi/N)$ 、 $V_2 = \exp(-j2\pi/N)$ の複素数で、 N, N はそれぞれ画像 $g(m, n)$ の横と縦の画素数

(13)

殆どない。変換後の画像の画素（画素）を適正なものにする一つの方法として、最近傍法という方法が知られている。その方法では座標点 (X_0, Y_0) (51)に最も近い格子点(52)の画素を選んで変換後の座標 (X_1, Y_1) の画素とする。計算は座標点 (X_0, Y_0) を四捨五入によって整数座標とすることによって行う。

この応用例に対する、本発明の適用について説明する。先ず、式(8)の係数 (b_{ij}) を本発明で使用する対数表現数値、すなわち式(1)の形に変換する。これは通常1回だけであるから高速度である必要はない。次に変換後の座標 (X_1, Y_1) （整数）を入力ルックアップテーブル(1)を使用して、対数表現数値に変換し（テーブル変換だから高速度）、式(8)を対数表現数値演算法利用の演算装置(2)で計算する（高速度高精度）。結果は指数に相当する部分が固定小数点数であるのでそれをアドレスとする出力ルックアップテーブル(3)により四捨五入して、整数値に変換する（高速度）。これがすなわち最近傍法の座標アドレス計算である

(12)

である。これについても、 V_1, V_2 のそれぞれを対数表現数値で与え、 $g(m, n)$ はあまり大きくない整数（例えば0-255）だとすると、実施例1と同じように高速度高精度に演算できる。この演算は通常高速度フーリエ変換で行うので、それが更に高速度になる。ただし、出力は出力ルックアップテーブル(3)により必要なら、整数値ではなく実数値（固定小数点数、浮動小数点数または、対数表現数値）の形で出力する。

実施例3

1次元のデータ、例えば、音声データ（入力はビット数ではなく、もっと大きい数、12-14ビットで扱うことが望ましい）についても当然、高速度高精度に演算できる。音声データなどの1次元データに対するフーリエ変換は式(10)により定義される。

$$G(k) = \sum_{n=0}^{N-1} g(n) V^n \quad (10)$$

ただし、 $V = \exp(-j2\pi/N)$ で N はデータの数である。これについても、実施例2と同じ理由で高速度高

(14)

度に計算できる。

実施例 4

本発明はコンピュータグラフィックスにおける図形描画においても、非常に効果的である。式(11)は中心が原点にあり、半径がRの円の式である。

$$x^2 + y^2 = R^2 \quad (11)$$

式(11)について説明する。この式に書き換えることができる。

$$y = \pm \sqrt{R^2 - x^2} \quad (12)$$

式(12)でyが負でない部分だけをとると式(12)は式(13)となる。

$$y = \sqrt{R^2 - x^2} \quad (13)$$

式(13)において、xの0からR/√2 (x=y)までの整数値に対して、yを計算し四捨五入して整数値まで浮めると、それが第6図で示すような8分円の描画点(61)である。後は、y=xの直線、x軸、y軸に対して対称な点をもとめて行けば、完全な円になる。式(13)の計算ではxの

(15)

ところである(1点プロットする毎に必ず)。

(78)はx=yであるかどうかを判定するところである。(79)はx=yの場合、点(x,y)をプロットするところである。

しかし、本発明によれば第6図のような8分円の描画アルゴリズムは第8図のようになる。非常に簡便である。

第8図について説明する。変数x、yは整数であり、r1, r2, y1は対数表現数値、すなわち、式(1)の形式の数値を扱う。Rは半径で実数である。第8図において、(81)は初期化を行うところである。半径Rを四捨五入して(R0(8))はRを四捨五入して整数にすることを意味する。yとし、xをゼロ、r1をlog2(R²)とする(r1は値としては、ほぼR²を表す)。(82)は終了かどうかを判定するところである。(83)は点(x,y)をプロット(描画)するところである。(84)はx座標を1つ進めるところである。(85)は変数xをルックアップテーブルT(入力ルックアップテーブル(1))により対数表現数値x1に変換するとこ

(17)

平方、乗算、更にR²-x²の平方根を計算しなければならない。従って、通常、浮動小数点演算で計算する。しかしながら、浮動小数点演算では平方(または乗算)や平方根の計算には大変な時間がかかる。実際には、第7図で示すような増分法と言われる8分円発生アルゴリズム等が開発されていて、整数の計算だけで済む様になっている。

第7図について説明する。この図に出てくる変数は全て整数である。第7図において、(71)は初期化をするところである。xをゼロ、yを半径R、dを3-2Rとする。(72)は終了かどうかを判定するところである。(73)は点(x,y)をプロット(描画)するところである。(74)は次のプロット点か右側または右側の下のどちらであるかを判定するところである。(75)は次のプロット点か右側の場合の処理で、そのように判別変数dを更新するところである。(76)は次のプロット点か右側の下の場合の処理で、判別変数dをそのように更新し、y座標を一つ進める(-1)とすることである。(77)はx座標を1つ進め

(16)

るである。(86)は√(R²-x²)の計算を対数表現数値演算法利用の演算回路(2)で行うところである。ここではR²は既に計算されていて、xの平方と、乗算と、R²-x²の平方根の計算を行う。xの平方については、x1の階数部を2倍するか、1ビット左シフトするだけである。乗算は式(8)の方法でルックアップテーブルを使って行う。R²-x²の平方根の計算は上記の乗算の結果(対数表現数値)を2で割るか1ビット右シフトするだけである。全て高速にできる。結果をy1とする。(87)は対数表現数値から整数値へ変換するところである。これはルックアップテーブルt(出力ルックアップテーブル(3))により四捨五入して行う。これも高速である。(88)はx=yであるかどうかを判定するところである。(89)はx=yの場合、点(x,y)をプロットするところである。

以上により第8図のアルゴリズムは第7図のアルゴリズムに比べ高速であることが分かる。

なお、第7図で示すアルゴリズムでは半径Rは整数(>0)でなければならないが、第8図のアル

(18)

ルゴリズムでは半径Rは整数である必要はない。
その点においても、本発明の方が優れている。

以上個別の実施例を示したが、これらは、異なる個別の実施例であって、高速高精度の演算が要求されるところでは、広範囲に本発明の実施が有効である。また、本発明はハードウェアで有効に実施できるが、特に微細回路での実施が特に有効である。

(発明の効果)

本発明により、既存の回路、装置、その他のデータ入力装置によるデータや、メモリアドレス、データ配列の確保等、固定小数点型のデータで表わされる数値に対して、高速高精度演算ができるので、特別の対数型アナログアンプ(4)や指数型アナログアンプ(7)が必要なく、既存のコンピュータ等のデジタル回路に本発明の演算装置を加えることにより処理を高速高精度にすることができる。

図1図2、3、4、5及びU8の入出力が固定小数点以外についての有効性について追加説明す

(19)

る。

入力または出力が浮動小数点数の場合、既存の浮動小数点型のデジタルシグナルプロセッサ等とのやり取りが高速簡便にできる。また、入力または出力が対数表現数値の場合は他の対数表現数値演算法利用の演算装置とのやり取りが高速簡便になる効果がある。

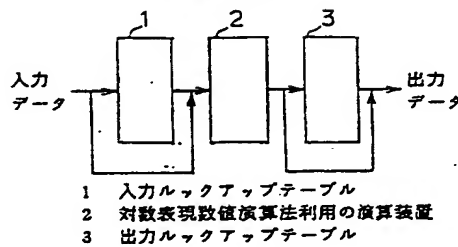
4. 図面の簡単な説明

第1図は本発明の構成図、第2図は対数表現数値演算法による従来の処理システムの構成図、第3図は対数表現数値の指数iとその表現値 a^i の関係図、第4図は数値の幾何学的交換の概念図、第5図は数値の幾何学的交換の計算座標(x_0, y_0)と格点の関係を示す図、第6図は8分円の描画直の図、第7図は増分法と言われる従来の8分円の描画アルゴリズムの図、第8図は本発明を適用した8分円の描画アルゴリズムの図。

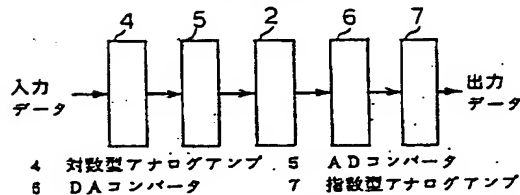
特許出願人 盛岡実夫

(20)

第1図



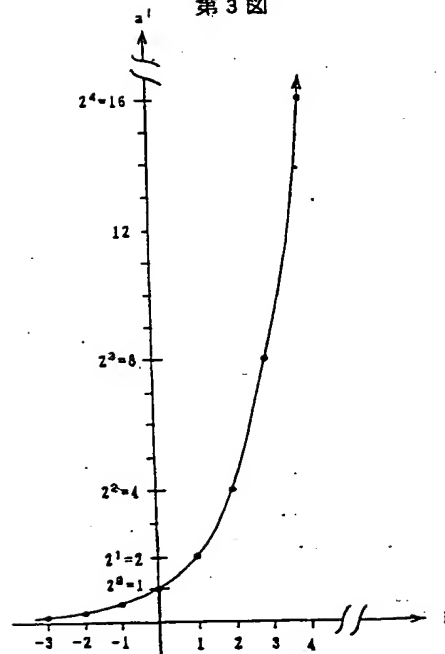
第2図



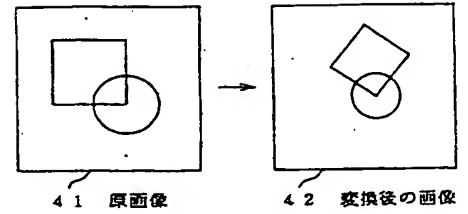
(7)

特開平 3-14128 (7)

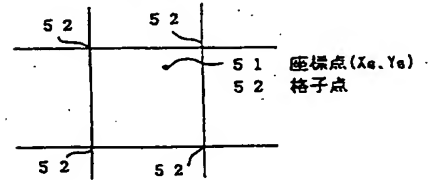
第 3 図



第 4 図



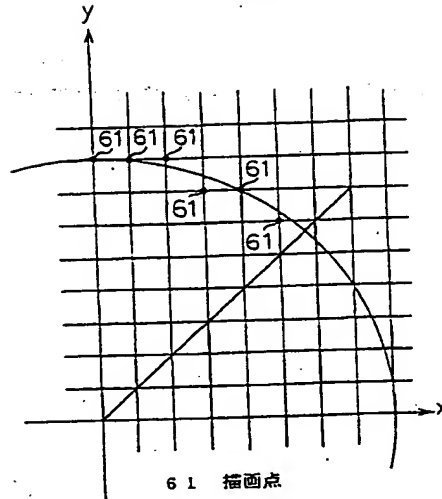
第 5 図



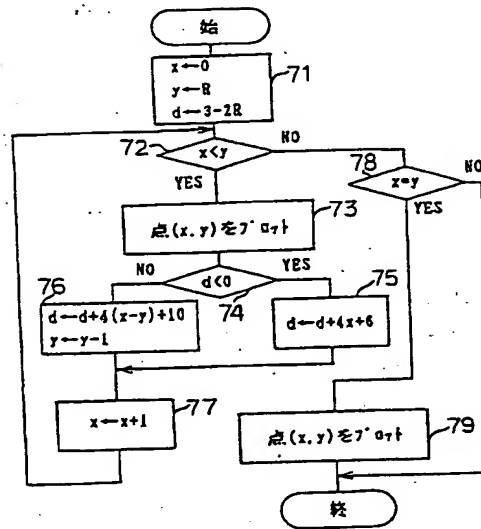
(8)

特開平 3-14128(8)

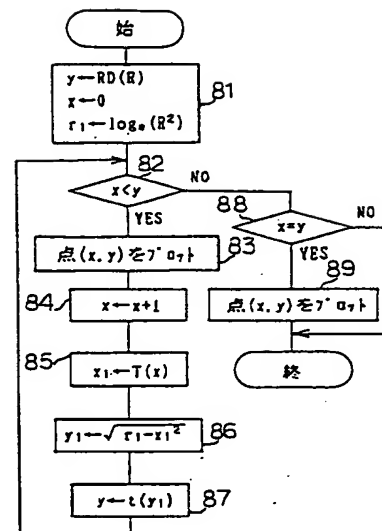
第 6 図



第 7 図



第 8 図



THIS PAGE BLANK (USPTO)